DISCOVERY

58(313), January, 2022

Simplified logic circuit design for efficient quantum computation with an Excess-3 Adder

Jeong Ryeol Choi✉[1], **Ji Nny Song**[2]

To Cite:
Choi JR, Song JN. Simplified logic circuit design for efficient quantum computation with an Excess-3 Adder. *Discovery*, 2022, 58(313), 58-62

Author Affiliation:
[1]Department of Nanoengineering, Kyonggi University, Gwanggyosan-ro, Yeongtong-gu, Suwon, Gyeonggi-do 16227, Republic of Korea
[2]Department of Nanoengineering, Kyonggi University, Gwanggyosan-ro, Yeongtong-gu, Suwon, Gyeonggi-do 16227, Republic of Korea

✉**Corresponding author:**
Department of Nanoengineering, Kyonggi University, Gwanggyosan-ro, Yeongtong-gu, Suwon, Gyeonggi-do 16227, Republic of Korea, Republic of Korea, e-mail: choiardor@hanmail.net

Peer-Review History
Received: 06 November 2021
Reviewed & Revised: 08/November/2021 to 09/December/2021
Accepted: 12 December 2021
Published: January 2022

Peer-Review Model
External peer-review was done through double-blind method.

DISCOVERY
SCIENTIFIC SOCIETY

ABSTRACT

Quantum computers are composed of quantum logic circuits that fulfill quantum computation based on quantum mechanics. However, during quantum computation, collapse of information in the circuits due to decoherence is a very subtle problem. In order to obtain exact computing results by preventing such decoherence, simplification of arithmetic logic circuits is important. In this work, we design a simplified Excess-3 Adder that can be available to decimal arithmetic operation. The simplification of the circuit is carried out by removing an operation line from the lately proposed circuit in this context. From a check of the logic process of arithmetic computations for numerical additions, we confirm that the circuit works well. Our circuit is robust from decoherence and consumes low electric power.

Keywords: Quantum computation, Excess-3 Adder, Decimal arithmetic circuit, Qubit

1. INTRODUCTION

Thanks to the rapid development of information technologies, primitive quantum computers have been realized now by several leading research groups (Arute et al., 2019; Zhong et al., 2020; Ball, 2021; Wu et al., 2021). Differently from classical computers that operate based on the principle of classical mechanics, the underlying mechanism of quantum computers is the principle of quantum mechanics. The main feature of quantum computation, which distinguishes it from existing computation, is that it adopts superposition and entanglement (Li and Yin, 2016). Both the superposition and entanglement are deeply related to quantum coherence, which is a natural but a weird characteristic of quantum mechanics. They are key resources of quantum information processing and its related quantum metrology.

Classical computers are inefficient in managing some information tasks. The examples of them are database searches and prime factorizations. However, such tasks can be carried out effectively by means of quantum computers. Grover algorithm (Morales et al., 2018) serves as a quantum protocol for database searches. Prime factorizations can also be accomplished by Shor's algorithm (Shor, 1997). Another defect of classical computers is that their logic circuits release lots of heat during arithmetic operation (Landauer, 1961). If the computation process is complicated, more heat is released. Contrary to this, quantum computation requires low electric power in general.

Logic circuits in computers are designed using a specific code. For this reason, decimal numbers should be converted to a required codeword. Although the binary code is the simplest one for that purpose, some decimal numbers cannot be represented in terms of the binary code (Thapliyal et al., 2007). Since this leads to errors in the computation, the binary code is inadequate as a code used for the purpose of exact computation. This is the reason why decimal codes are necessary in the process of computation. Some of well-known decimal codes are Excess-3 code, BCD (8421 code), 6311 code, 2421 code, EBCDIC, and so on (Choi and Song, 2019; Yeon et al., 2012; Blatov and Chudov, 1978; Kumar and Umadevi, 2015). Among them, we will adopt Excess-3 code in this work. In order to obtain a codeword of Excess-3 code, one should add 0011 in the codeword of the corresponding BCD. A merit of Excess-3 code in computing is self-complementing property (Kumar and Umadevi, 2015). In order to understand this property, let us see a 1’s complement of a codeword of Excess-3. For example, 1’s complement of a number 0100 (1 in decimal number) is 1011 (8 in decimal number). From this, we confirm that one can obtain 9(decimal)’s complement of a codeword by simply reversing each bit, i.e., by converting each bit as 1→0 and 0→1. This property can be efficiently used in the related arithmetic processes. Hence, Excess-3 code is convenient in computations, especially in subtraction.

A possible trouble in quantum computation is the occurrence of decoherence during its process (Duan and Guo, 1998). Decoherence is a main factor that hampers the development of quantum computers. The difficulty in manufacturing high-fidelity quantum computers with a large number of qubits is mainly due to this. Hence, a simplification of computing logic circuits may reduce such decoherence. Regarding this, we will simplify the existing decimal arithmetic circuit based on Excess-3 code in this work.

2. MATERIALS AND METHODS

The Excess-3 Adder can be constructed using full and half Adders. Although the computing process of the quantum full and half Adders gives the same results as that of the classical full and half Adders, they work according to the principle of quantum mechanics. The detailed flowchart for the process of decimal addition with the Excess-3 code is given in Ref. (Yeon et al., 2012). Based on that chart, the circuit of the Excess-3 decimal Adder will be constructed rigorously. In this design, we will consider the minimization of necessary operations. It will also be tested whether the circuit works without errors.

3. RESULTS AND DISCUSSION

3.1. Design of the Excess-3 Adder

In order to minimize decoherence during quantum computing processes, simplification of logic circuits is necessary. A method for such simplification regarding the operation process is reducing the number of operation lines (qubits). The circuit that we suggest for this purpose is shown in Fig. 1. This circuit is designed on the basis of Excess-3 code. The process of decimal addition in this circuit is represented as ($A_4A_3A_2A_1$) + ($B_4B_3B_2B_1$) + input carry[C_i] = ($S_4S_3S_2S_1$) + output carry[C_o]. This circuit can be used as a basic element of computing processes in quantum computers. The addition process adopted in this circuit can also be extended to other computing processes, such as subtraction, multiplication, and division.

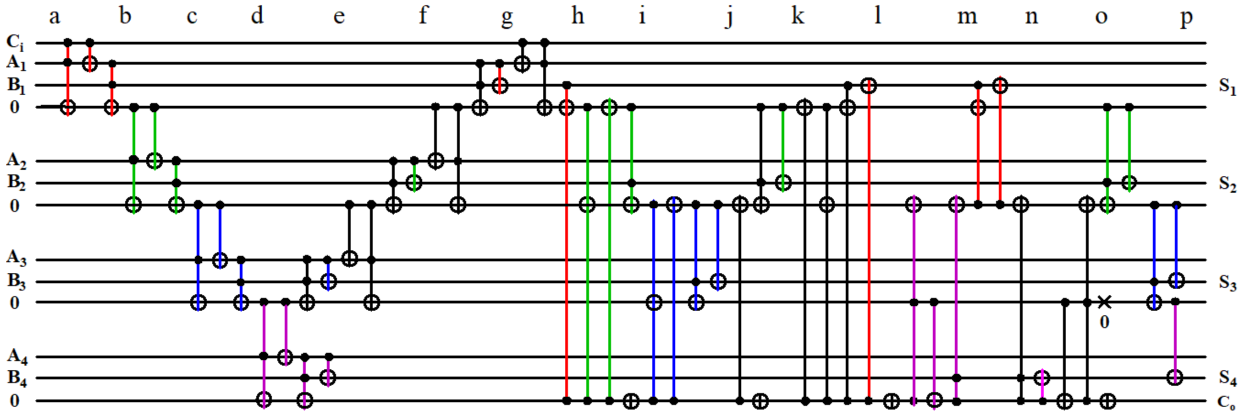

Figure 1. Diagram of simplified Excess-3 Adder

The symbol X in the 10th operation line in Fig. 1 means resetting of the qubit to zero (Thapliyal and Ranganathan, 2010). The number of operation lines in this circuit is 13, while that in the latest previous design in this context, given in Ref. (Choi and Song, 2019), is 14. Hence, the operation lines in our circuit are less than 1 compared to those of the previous design. Among 13 operation lines in our circuit, 3 lines are garbage lines. In this design, all three garbage lines start with an initial value 0. Apparently, the operation process in this circuit is simpler than that of Ref. (Choi and Song, 2019). This is the main contribution of this work in the circuit design.

Table 1. The change of qubit data over time for the process of computation supposed in Example 1.

	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	
C_i	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
A_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
B_1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	S_1
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
A_2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
B_2	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	S_2
	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0	
A_3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
B_3	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	S_3
	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	
A_4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
B_4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	S_4
	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	0	C_o

Table 2. The change of qubit data over time for the process of computation supposed in Example 2.

	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	
C_i	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
A_1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
B_1	1	1	1	1	1	1	0	0	0	0	0	1	1	1	1	1	S_1
	0	1	1	1	1	1	0	0	1	1	1	0	0	0	0	0	
A_2	1	1	0	0	0	0	1	1	1	1	1	1	1	1	1	1	
B_2	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	S_2
	0	0	1	1	1	1	0	0	1	1	0	0	0	0	0	0	
A_3	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	
B_3	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	S_3
	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	
A_4	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	
B_4	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	S_4
	0	0	0	0	1	1	1	1	0	0	1	1	0	0	0	1	C_o

3.2. Testing of the operation process

Here we check whether the logic circuit given in Fig. 1 works well by introducing two examples of the computing process now:

3.2.1. Example 1: Let us first see the case where there is no carry in the computation result. For this purpose, we will see the operation process which is 3+5=8 (in Excess-3 code: 0110 + 1000 = 1011).

3.2.2. Example 2: Now we check the case of computation 4+8=12 (0111 + 1011 = 0100 0101) of which result involves a carry.

We suppose that there is no input carry for both examples for simplicity, i.e., C_i =0.

From table 1, we see that the result of Example 1 is ($S_4S_3S_2S_1$) = (1011), whereas table 2 shows that the result of Example 2 is ($S_4S_3S_2S_1$) = (0101) with an output carry. Thus, we confirm that the two cases produce exact results.

4. CONCLUSIONS

Through the construction of logic circuits based on quantum principle, it may be possible to carry out high-speed computations in some tasks. However, preventing the collapse of quantum states in the circuits by decoherence induced by the environment and/or complicated operating processes is a challenging problem (Brune et al., 1996). While quantum coherence is necessary to perform quantum computing, a loss of coherence results in the disappearance of quantumness of the computing system. For this reason, we cannot neglect decoherence in quantum circuits. Hence, the simplification of logic circuits is crucial for efficient computations.

According to the above requirement, a simplified decimal arithmetic circuit has been designed using Excess-3 code. The Excess-3 Adder we proposed here is composed of 13 operation lines, where 3 lines among them are garbage lines. On the other hand, the circuit design in the previous work (Choi and Song, 2019) is composed of 14 lines. We confirmed that our circuit works well without loss of functional facilities. The simplification of the quantum logic circuit in this work may help to resolve the problem of decoherence. In addition, the circuit we have proposed may consume low electric power.

Authors contributions

Both authors contributed equally.

Funding

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No.: NRF-2021R1F1A1062849).

Conflicts of interests

The authors declare that there are no conflicts of interests.

Data and materials availability

All data associated with this study are present in the paper.

REFERENCES AND NOTES

1. Arute, F. et al. 2019. Quantum supremacy using a programmable superconducting processor. Nature 574(7779): 505-510.
2. Ball, P. 2021. First 100-qubit quantum computer enters crowed race. Nature 599(7886): 542.
3. Blatov, V.V. and Chudov, A.A. 1978. Binary-decimal adder-subtractors. Instrum. Exp. Tech. 21(5 pt 1): 1260-1264.
4. Brune, M.E., Dreyer, H.J., Maître, X., Maali, A., Wunderlich, C., Raimond, J.M. and Haroche, S. 1996. Observing the progressive decoherence of the "meter" in a quantum measurement. Phys. Rev. Lett. 77(24): 4887-4890.
5. Choi, J.R. and Song, J.N. 2019. Improved design and optimization of Excess-3 Adder for quantum computation. Discovery 55(281): 210-213.
6. Duan, L.-M. and Guo, G.-C. 1998. Reducing decoherence in quantum-computer memory with all quantum bits coupling to the same environment. Phys. Rev. A 57(2): 737-741.
7. Kumar, H.A.A. and Umadevi, S. 2015. Implementation of fast radix-10 BCD multiplier in FPGA. Indian J. Sci. Technol. 8(19): 1-6.
8. Landauer, R. 1961. Irreversibility and heat generation in the computing process. IBM J. Res. Develop. 5(3): 183-191.
9. Li, T. and Yin, Z.-Q. 2016. Quantum superposition, entanglement, and state teleportation of a microorganism on an electromechanical oscillator. Sci. Bull. 61(2): 163-171.
10. Morales, M.E.S., Tlyachev, T. and Biamonte, J. 2018. Variational learning of Grover's quantum search algorithm. Phys. Rev. A 98(6): 062333.
11. Shor, P.W. 1997. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum

computer. Siam J. Comput. 26(5): 1484-1509.

12. Thapliyal, H., Arabnia, H.R., Bajpai, R. and Sharma, K.K. Partial reversible gates(PRG) for reversible BCD arithmetic. Proceedings of the 2007 International Conference on Computer Design(CDES'07), Las Vegas, USA, June 2007, pp. 90-91 (CSREA Press).
13. Thapliyal, H. and Ranganathan, N. 2010. Design of reversible sequential circuits optimizing quantum cost, delay, and garbage outputs. ACM J. Emerg. Technol. Comput. Syst. 6(4): 14.
14. Wu, Y. et al. 2021. Strong quantum computational advantage using a superconducting quantum processor. Phys. Rev. Lett. 127(18): 180501.
15. Yeon, K.H., Choi, J.R., Kim, D., Kim, M.-S. and Maamache, M. 2012. Reversible quantum computation using Excess-3 code. AIP Conf. Proc. 1444: 310-313.
16. Zhong, H.-S. et al. 2020. Quantum computational advantage using photons. Science 370(6523): 1460-1463.